

設計書

設計内容の概要

- ユーザーは、LINE メッセージで目覚ましアラームを「MMDDhhmm」形式で指定する。スヌーズ機能や曜日指定にも対応し、アラームの有効・無効を切り替えることも可能。
- Remo 3 のセンサと外部 API（気象庁・天文台 API）を利用し、以下の情報を毎日取得：
 1. 日の出・日没時刻
 2. 天気（晴れ・曇り・雨）
 3. 室内の照度（Remo 3 センサより）
- 取得情報に基づいて以下の制御を行う：
 1. 日の出時刻：目覚まし時刻の設定がない場合照明 ON
 2. 日没 1 時間前：ユーザーが在宅の場合のみ照明 ON
 3. 天気による日の出及び起床時に点灯する照明色変化（晴れ＝白、曇り＝青白、雨＝黄）
 4. 室内の輝度に応じて照明の明るさ自動調節
- ユーザーは単純な電気のオンオフ機能や明るさ調整も行うことができる。

システム処理の流れ

- 日次初期化処理（午前 0 時に実行）
 1. 日次初期化処理（午前 0 時に実行）
 2. 天気 API から天気を取得
 3. スケジュール登録（Google Calendar または内部配列）
- センサ定期チェック（5 分おき）
 1. Remo 3 から照度取得
 2. 明るさに応じて照明 ON/OFF または調光を実行
- 目覚まし制御
 1. LINE から指定された時刻を受信・解析
 2. 指定時刻にアラーム動作（照明 ON または音源操作）
 3. スヌーズ（例：5 分後再実行）も可能
 4. 曜日指定時はその曜日にのみ実行
- 天気と照明色の連動
 1. 天気 API から情報取得
 2. 照明の色をプリセットに応じて変更（Remo 登録済み赤外線信号で実現）
- 日没時の照明点灯

1. 日没 1 時間前に人感センサーまたは在宅フラグを確認
2. 在宅なら照明点灯、いなければスキップ

必要なモジュール（.gs ファイル）

1. **line_handler.gs**
 - LINE Webhook の受信
 - ユーザーからのコマンド（MMDDhhmm 形式など）の解析
 - スケジュールデータの登録・管理
2. **schedule_manager.gs**
 - 目覚ましスケジュールの登録と実行
 - 日の出・日没イベントの管理
 - 曜日指定やスヌーズ対応など
3. **sensor_reader.gs**
 - Nature Remo 3 からの照度センサーデータ取得
 - 室内明るさに応じた照明レベルの判定
4. **weather_fetcher.gs**
 - 気象庁 API や OpenWeatherMap からの天気情報取得
 - 天気データの解析（晴れ・曇り・雨の判定）
5. **alarm_controller.gs**
 - アラーム時の照明 ON や赤外線信号の送信
 - スヌーズ機能の実装
6. **light_controller.gs**
 - 天気や照度、日の出・日没時間に応じた照明制御
 - 照明の色や明るさの変更処理